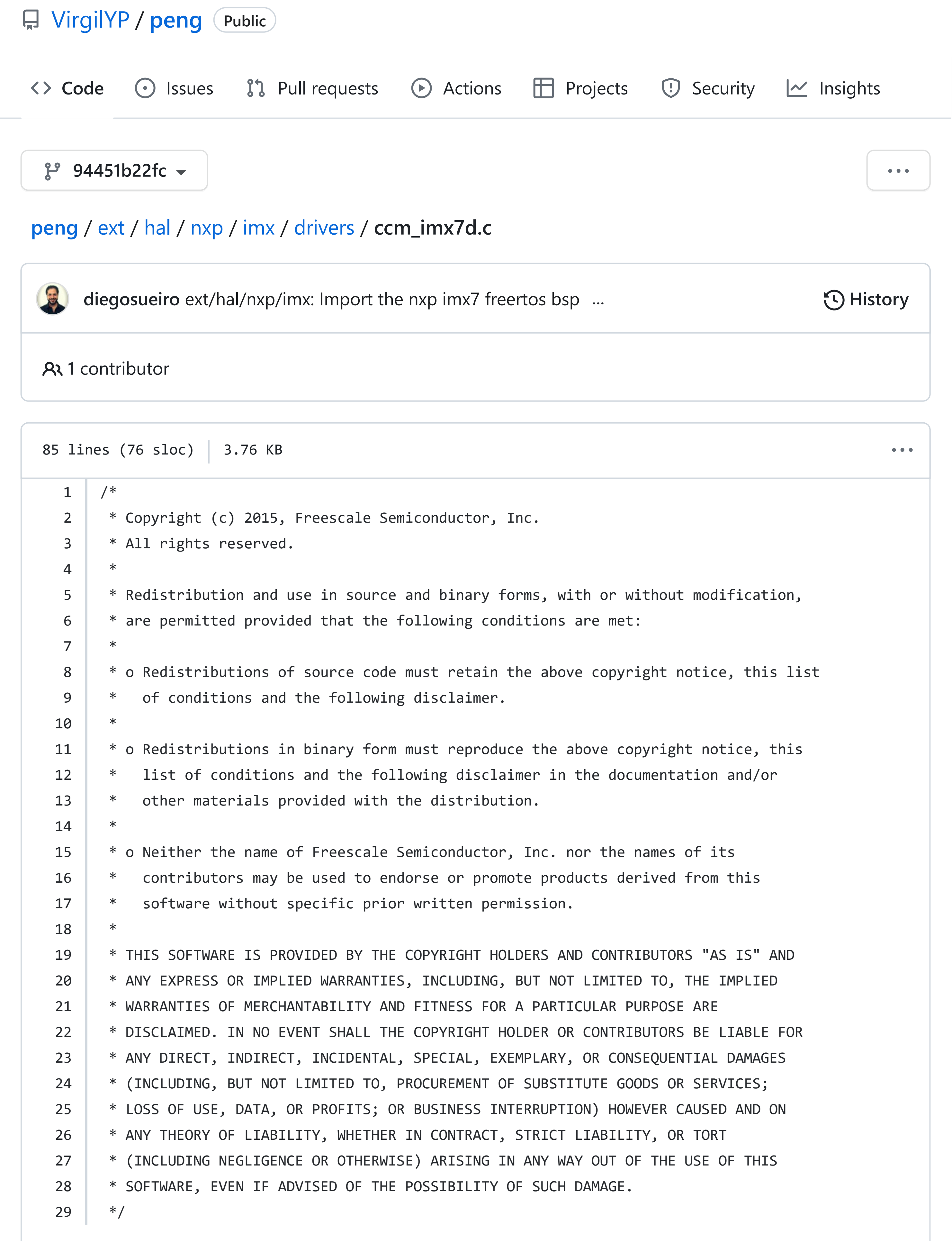
EXHIBIT B19



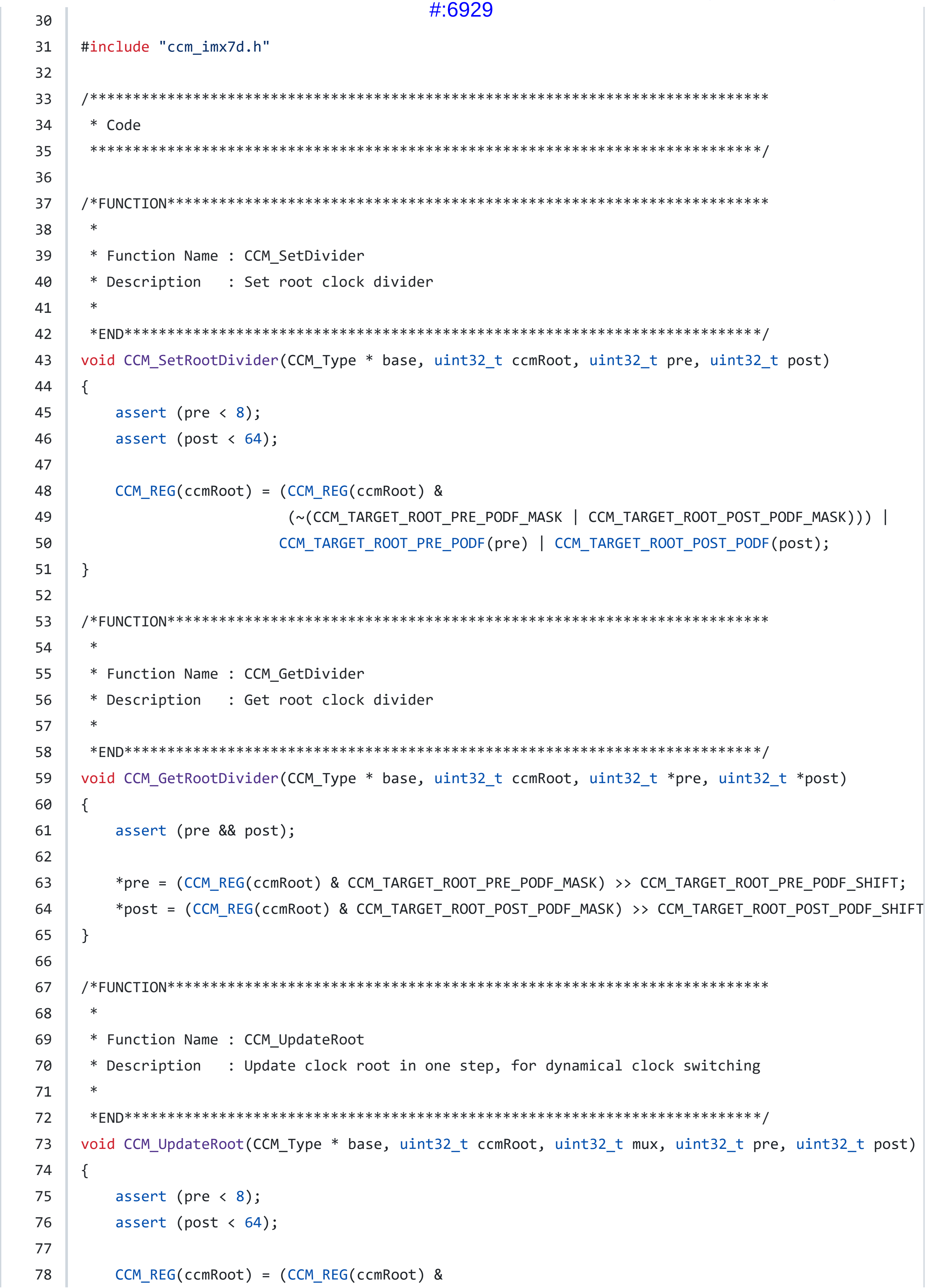
peng / ext / hal / nxp / imx / drivers / ccm_imx7d.c

diegosueiro ext/hal/nxp/imx: Import the nxp imx7 freertos bsp …

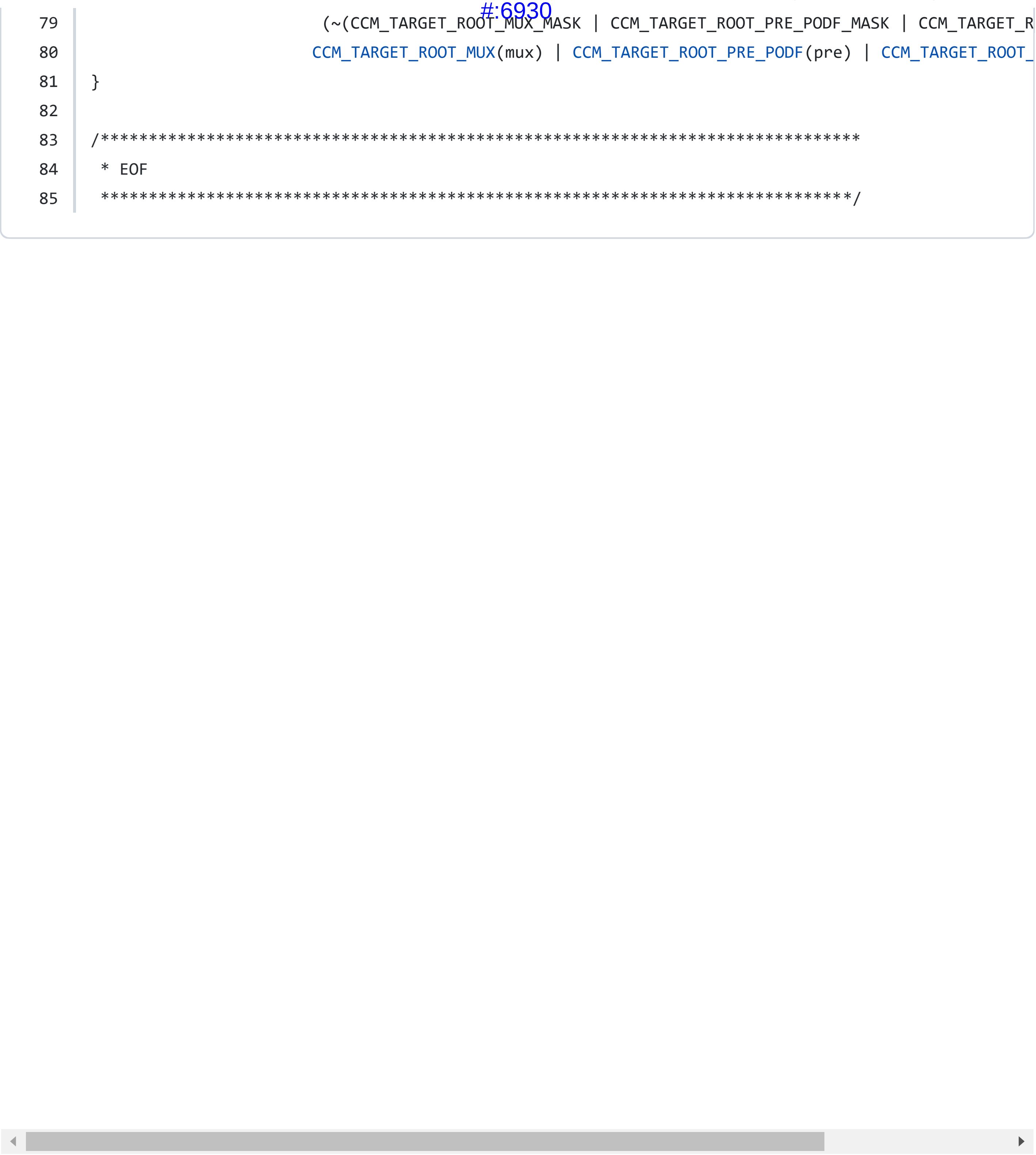
1 contributor

85 lines (76 sloc) | 3.76 KB

```
/*
 * Copyright (c) 2015, Freescale Semiconductor, Inc.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without modification,
 * are permitted provided that the following conditions are met:
 *
 * o Redistributions of source code must retain the above copyright notice, this list
 *   of conditions and the following disclaimer.
 *
 * o Redistributions in binary form must reproduce the above copyright notice, this
 *   list of conditions and the following disclaimer in the documentation and/or
 *   other materials provided with the distribution.
 *
 * o Neither the name of Freescale Semiconductor, Inc. nor the names of its
 *   contributors may be used to endorse or promote products derived from this
 *   software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
 * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 * DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR
 * ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
 * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
 * ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
 * SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */
```



```

#include "ccm_imx7d.h"

/*******************************************************************************
 * Code
 ******************************************************************************/

/*FUNCTION**********************************************************************
 *
 * Function Name : CCM_SetDivider
 * Description   : Set root clock divider
 *
 *END**************************************************************************/
void CCM_SetRootDivider(CCM_Type * base, uint32_t ccmRoot, uint32_t pre, uint32_t post)
{
    assert (pre < 8);
    assert (post < 64);

    CCM_REG(ccmRoot) = (CCM_REG(ccmRoot) &
                        (~(CCM_TARGET_ROOT_PRE_PODF_MASK | CCM_TARGET_ROOT_POST_PODF_MASK))) |
                       CCM_TARGET_ROOT_PRE_PODF(pre) | CCM_TARGET_ROOT_POST_PODF(post);
}

/*FUNCTION**********************************************************************
 *
 * Function Name : CCM_GetDivider
 * Description   : Get root clock divider
 *
 *END**************************************************************************/
void CCM_GetRootDivider(CCM_Type * base, uint32_t ccmRoot, uint32_t *pre, uint32_t *post)
{
    assert (pre && post);

    *pre = (CCM_REG(ccmRoot) & CCM_TARGET_ROOT_PRE_PODF_MASK) >> CCM_TARGET_ROOT_PRE_PODF_SHIFT;
    *post = (CCM_REG(ccmRoot) & CCM_TARGET_ROOT_POST_PODF_MASK) >> CCM_TARGET_ROOT_POST_PODF_SHIFT
}

/*FUNCTION**********************************************************************
 *
 * Function Name : CCM_UpdateRoot
 * Description   : Update clock root in one step, for dynamical clock switching
 *
 *END**************************************************************************/
void CCM_UpdateRoot(CCM_Type * base, uint32_t ccmRoot, uint32_t mux, uint32_t pre, uint32_t post)
{
    assert (pre < 8);
    assert (post < 64);

    CCM_REG(ccmRoot) = (CCM_REG(ccmRoot) &
```



```
                        (~(CCM_TARGET_ROOT_MUX_MASK | CCM_TARGET_ROOT_PRE_PODF_MASK | CCM_TARGET_R
                       CCM_TARGET_ROOT_MUX(mux) | CCM_TARGET_ROOT_PRE_PODF(pre) | CCM_TARGET_ROOT_
}

/******************************************************************************
 * EOF
 *****************************************************************************/
```